Amendments to the Claims:

Please cancel Claim 2 without prejudice.

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) An In-Circuit Emulation system, comprising:

a microcontroller having a microcontroller clock;

a virtual microcontroller running in lock-step synchronization with the microcontroller by executing the same instructions using the same clocking signals between said microcontroller and said virtual microcontroller;

a gatekeeper circuit coupled to the virtual microcontroller and the microcontroller; ~~and~~

a gatekeeper clock running independent of the microcontroller clock to clock operations carried out in the gatekeeper circuit; and

a host computer running In-Circuit Emulation debug software, the host computer being in communication with the gatekeeper circuit so that halt commands and requests for data from the virtual microcontroller are passed through and regulated by the gatekeeper circuit to assure that emulator operations are not disrupted.

2. (canceled).

3. (original) The apparatus according to Claim 1, wherein the gatekeeper circuit comprises means for determining that the microcontroller is in a sleep state.

4. (previously presented) The apparatus according to Claim 1, wherein the gatekeeper circuit determines that the microcontroller is in the sleep state by determining if the microcontroller clock is operating.

5. (previously presented) The apparatus according to Claim 1, wherein the gatekeeper circuit determines that the microcontroller is in the sleep state by determining if the microcontroller clock is operating and a data line from the microcontroller is in a prescribed logic state.

6. (original) The apparatus according to Claim 3, wherein the gatekeeper circuit further comprises means for notifying the host computer of the microcontroller's state in the event the microcontroller is in a sleep state.

7. (previously presented) The apparatus according to Claim 1, wherein the gatekeeper further comprises means for receiving a halt command

from the host computer and for queuing a break to the microcontroller and the virtual microcontroller in response thereto.

8. (original) The apparatus according to Claim 7, wherein the gatekeeper further comprises means for detecting that a break has occurred in the microcontroller and the virtual microcontroller and for notifying the host computer that the break has occurred.

9. (original) The apparatus according to Claim 7, wherein the halt command comprises one of a programmed breakpoint and a user initiated manual halt command.

10. (original) The apparatus according to Claim 7, wherein the halt command is issued by a breakpoint controller in response to detection of a programmed breakpoint.

11. (original) The apparatus according to Claim 1, wherein the gatekeeper further comprises means for permitting access to registers and memory locations in the virtual microcontroller when the microcontroller and the virtual microcontroller are in a halted state.

12. (original) The apparatus according to Claim 1, wherein the halt command comprises a user initiated manual halt command.

13. (currently amended) A method of regulating a host computer's access to a virtual microcontroller, operating in lock-step synchronization with a ~~real~~ microcontroller by executing the same instructions using the same clocking signals between said microcontroller and said virtual microcontroller, ~~and~~ by using a gatekeeper function to assure that emulator operations are not disrupted, said method comprising:

running said gatekeeper clock independent of the microcontroller clock to clock operations;

receiving a halt command;

queuing a break command to the microcontroller and the virtual microcontroller in response to the halt command; and

upon execution of the break command, permitting the host computer to have access to registers and memory locations in the virtual microcontroller.

14. (original) The method according to Claim 13, wherein the halt command is received as a user initiated manual halt command from the host computer.

15. (original) The method according to Claim 13, wherein the halt command is received from breakpoint controller to initiate a programmed breakpoint.

16. (original) The method according to Claim 13, further comprising determining if the microcontroller and the virtual microcontroller are in a sleep state upon receipt of the halt command.

17. (previously presented) The method according to Claim 16, wherein said gatekeeper determines that the microcontroller is in the sleep state is carried out by determining if a microcontroller clock is operating.

18. (original) The method according to Claim 17, wherein determining that the microcontroller is in the sleep state is carried out by determining if a microcontroller clock is operating and a data line from the microcontroller is in a prescribed logic state.

19. (previously presented) The method according to Claim 16, further comprising notifying the host computer of the microcontroller's state in the event the microcontroller is in the sleep state.

20. (original) The method according to Claim 13, further comprising notifying the host computer when the microcontroller and the virtual microcontroller are halted.

21. (currently amended) A method of regulating a host computer's access to a virtual microcontroller operating in lock-step synchronization with a ~~real~~ microcontroller by executing the same instructions using the same clocking signals between said microcontroller and said virtual microcontroller, ~~and~~ by using a gatekeeper function to assure that emulator operations are not disrupted, said method comprising:

running said gatekeeper clock independent of the microcontroller clock to clock operations;

receiving a halt command as one of a user initiated manual halt command from the host computer and a breakpoint controller initiated halt command for a programmed breakpoint;

determining that the microcontroller is in the sleep state is carried out by determining if a microcontroller clock is operating and a data line from the microcontroller is in a prescribed logic state;

notifying the host computer of the microcontroller's state in the event the microcontroller is in the sleep state;

queuing a break command to the microcontroller and the virtual microcontroller in response to the halt command;

notifying the host computer when the microcontroller and the virtual microcontroller are halted; and

upon execution of the break command, permitting the host computer to have access to registers and memory locations in the virtual microcontroller.